$2.00

THE BBC MICROCOMPUTER USERS GROUP OF NZ

NEWSLETTER

Contents:

AGM notes 2
PROClibrary - PROCdiagonal 2
- PROCoverlay 2
BEEBUG's Mini text editor 3
CHAINS - a must for Model A's 6
For, By, To the Editor 7
Synchronising sounds 8
Towers of Hanoi - game listing 8
Which Whitcoulls? 11
User-defined keys 12
Extended Printer character set 15
Tape tip 15
Sketch logic rewritten 16
What others are doing 16
Prime numbers 17
Penny-pinching Printer Project Pt.1 18
Technical Aids for Handicapped people 19
Polar Plotting 20
On filling the newsletter 23
Discounts, deals... 24
Meetings & Memberships 24

ABS(COS(9*A)*COS(3*A))

VOLUME 1 NUMBER 5 MAY 1983

AGM

This is just some brief notes on the first AGM of the BBC Microcomputer Users Group of NZ held last month. It is not the minutes. There were 35 adults attending the AGM, plus several younger members who preferred to enjoy the games software provided on several machines around the room.

The President's report covered a brief history of the Users Group since the first meeting early in December 1982, and how much ground had been covered since then. During the Treasurer's report, a member from the 'audience' noted that we were only financial because we had stock of tapes & magazines.

CONSTITUTION

A draft of the proposed Constitution had been sent to the Registrar of Incorporated Societies seeking their comments on it. Members felt that when their response had been received, and alterations made, the Constitution should then be circulated to all members for their comments (with a deadline!) before preparing the final version.
There was some discussion on Membership fees - the opinion voiced by some was that perhaps we were setting them too low! However, cheaper postal rates for the magazine are being investigated, which should increase the amount available for other member services.

One of the Wellington retailers was prepared to be particularly helpful in various ways, including standing on the Committee. However, if we are to maintain our image of independence, the National body cannot be seen to be too close to any retailer.

WELLINGTON BRANCH:

A brief meeting after the national AGM set up a small band of people willing to help organise monthly meetings in Wellington, and also set a levy of $5.00 over and above the National membership, to cover room hire, supper etc.

FUNCTION & PROCEDURE LIBRARY

Jeff (in Alexandra) has found the ability to call a PROCedure directly, in immediate mode useful for debugging etc., when the DEF PROC is within a listing. For example, enter this listing:

```
10 DEF PROCdiagonal(A$,X%,Y%)
20 CLS
30 FOR A%=1 TO LEN(A$)
40   PRINTTAB(X%,Y%)MID$(A$,A%,1)
50   X%=X%+1:Y%=Y%+1
60   NEXT
70 ENDPROC
```

```
100 DEF PROCoverlay(A$,X%,Y%)
110 VDU22,5,5
120 MOVE X%,Y%:GCOL0,2:PRINTA$
130 MOVE X%+4,Y%+4:GCOL0,1:PRINTA$
140 VDU4
150 ENDPROC
```

Now enter `PROCdiagonal("Whodunnit?",5,5)` and the text will be printed diagonally across and down the screen. Change line 40 to:

```
     40 Y% = Y%+1
```

and it will be printed vertically.

PROCoverlay is a little trick for overlaying text to give a 3-D effect.
So, enter `PROCoverlay("GUESS who!",100,500)`

NOTES: 1. PROCoverlay prints text in graphics mode which allows the greater accuracy in placing the characters on the screen than PRINT TAB.
2. See previous articles on VDU22 - if you have any problems, enter MODE 5 <RETURN> before calling these PROCs.

MINI TEXT EDITOR VERSION II+

David GRAHAM

Reproduced with the kind permission of BEEBUG magazine

Our thanks are given to the Editors of Beebug magazine for permission to reproduce this program and article in our newsletter. They have also supplied us with some enhancements beyond those printed in their November 1982 issue. We have incorporated them in the program, and renumbered it. It is designed for a 32K machine, but by changing the Mode in lines 25010 and 25250 to Mode 7 it will work in a 16K machine.

This is a minute wordprocessor substitute. It uses the Beeb's own Basic text editing facilities to create and edit files of text. These are entered as lines of Basic using the AUTO command, and may be easily saved, loaded, edited and printed out. The program does not allow right-justification or closing up of text, so inserting words could cause a line to overflow - in which case the neatest thing to do may be to retype the paragraph.

The function keys are used extensively in this editor, and a card with their definitions should be used. Type CHAIN"" to load the program, followed by key f1 or f2. Either key will initiate the AUTO mode for text entry, but f2 will give your address as a letter head (In the listing given, BEEBUG's address is used - substitute your own at lines 1010,1020 etc.). Alternatively, pressing f1 will erase the address from the machine.

Then simply type in the text you require, using f9 when you reach the end of a line. If you go over the end of a line, use the 'delete' key to reduce the line length. As the keys stand, f6 will tab along 6 spaces, and f8 tabs to the right side for entering addresses. Key f3 is unprogrammed, and may be programmed with a word or short sentence that you use often.

When you have finished entering text, press Escape, then use the LIST command, and editing keys. Pressing f7 will return Mode 7 for better visibility. You can return to Mode 3 with f0, which just executes RUN. To return to entering text after you have escaped from the line numbering mode, you must type in AUTOx where x is greater than the number of the last Basic line that holds previously-entered text. For this reason, it is easier to correct mistakes at the end rather than as you go along.

To save the text created, use the SAVE"name" command, and save and load as a normal program. This means that each text file saved, conveniently has a copy of the editor with it.

Two modes of printout are available. f5 gives a printout with Basic line numbers, for ease of subsequent editing; while f4 calls a routine which ignores the line numbers. A number of possible options are available with this form of printout, and those chosen will depend on the paper feed mechanism of your printer, and the number of lines required. As the program stands, it will send a form feed character every 31 lines of text (check your manual to see whether your printer responds to this). This may be useful if you are printing double-spaced text (achieved by typing *FX6,0 - assuming that this command has not already been used). If you require a different number of lines printed before a form feed, alter line 25430 of the program, changing the value from 31 to that you require. If you do not want the form feed at all, delete line 25430. If you want the program to stop after say, 50 lines of text have been printed, so you can insert another sheet of paper, alter line 25430 to read:

```
  IF L%>50 THEN wait=GET:L%=0
```

Pressing any key will cause another page to be printed.

To increase the spacing between lines, alter line 25410 IF ?K%=13 VDU1,10
This may be extended to VDU1,10,1,10 or VDU1,10,1,10,1,10 etc to increase the spacing further. Each "1,10" in the VDU command adds an extra line feed after each line of text.
Note that as the program stands, it can give triple spacing on some printers. (eg the C.ITOH). To cancel these, delete lines 25410 AND 25440. This returns you to single spacing for those printers.

Incidentally, when you select printout without line numbers using f4, you are asked to enter a continuation line number. This is just to allow you to start printout at any line of a page. As the program is here, if you enter 29 here, the program will print out 2 lines of text before doing a form feed. For the benefit of printers using a continuous roll of paper (and for spacing down the letter head), printout is prefaced with a number of blank lines. These may be removed by deleting lines 950 to 990. At the end of the printout the screen will display a word count and a lines in last page count.

THE ASTERISKS: the original program could not accept Basic reserved words as text without corrupting them. eg P.O.Box would appear on numbered printout as PRINTOLDBox or worse on unnumbered printout. As the keys are defined, each line begins with an asterisk. The BASIC tokenising routine then considers this to be a MOS command and switches off the tokenising routine. If you press RETURN instead of f9, enter a * at the start of the new line.

If you are using the serial interface (RS423) for your printer, you will need to customise your program in this way: First of all change line 2 to read serial = TRUE . This sets the program for serial output at 1200 baud. If your printer requires a different baud rate, replace line 5 with the appropriate command:

*FX8,1	75 baud	*FX8,5	2400 baud
*FX8,2	150 baud	*FX8,6	4800 baud
*FX8,3	300 baud	*FX8,7	9600 baud
*FX8,4	1200 baud	*FX8,8	19200 baud

All printers behave in different ways, and word processors costing £50-plus are or can be, customised for a given printer. This is obviously not possible in this case. Although we have tried to provide for a number of different printer configurations, we cannot cover them all, and would urge you to experiment with the program to achieve the results you require.

NOTE
If you are modifying the program, leave lines 1-10 exactly intact in order to retain the correct functioning of the printout routine.

Adding 25485 VDU2,1,12,3 will give a form feed at the end of your text on an Epson printer.

If you use this editor on a disc-based system, alter line 25270 N%=PAGE+1
This change only affects the printout without line numbers routine, and does not alter its correct operation on a cassette-based system.

```
    1 REM MINI TEXT EDITOR
    2 serial=FALSE
    3 IF NOT serial THEN 9
    4 *FX5,2
    5 *FX8,4
    9 GOTO25000
   10
  950
  960
  970
  980
  990
 1010 *
 1020 *
 1030 *
 1040 *
25000 REM KEY SET ROUTINE
25010 MODE3
25020 *KEY0RUN|M
25030 *KEY1DELETE1000.1100|MAUTO1000.10|M*
25040 *KEY2L.1000,1099|MAUTO1100|M*
25050 *KEY4GOTO25240|M
25060 *KEY5VDU2|MLIST100.10000|MVDU3|M"
25070 *KEY6"          "
25080 *KEY 7 "MO.7 |M"
25090 *KEY 8 "
25100 *KEY9|M*
25110 PROCWINDOW
25120 END
25130 DEF PROCWINDOW
25140 S$="I          "
25150 PRINTTAB(8,23)S$S$S$S$S$S$S$"I
"
25160 VDU28,0,22,79,0
25170 PRINTTAB(0,18):"B E E B U G  M
ini Text Editor"
25180 PRINT"(printout routine by Ian
 Sinclair)"
25190 PRINTTAB(6,22):
25200 PRINT"PRESS KEY F1 OR F2 TO ST
ART (F2 PRINTS ADDRESS)"
25210 PRINT''"OR F4 OR F5 FOR PRINTOU
T"
25220 PRINT''''
25230 ENDPROC
25240 REM PRINTOUT WITHOUT NUMBERS
25250 MODE3:PRINT'''
25260 PRINTTAB(2)"Please choose cont
inuation or new print- type line num
ber or zero.";:INPUT A$:L%=VAL(A$):W
%=0
25270 N%=3585
25280 VDU 2
25290 REPEAT
25300   N%=N%+?(N%+2)
25310   UNTIL?(N%+1)=10
25320 J%=N%+2
25330 REPEAT
25340   L%=L%+1
25350   FORM%=1TO6
25360     PRINTCHR$(1)" ";
25370     NEXT
25380   FOR K%= J%+2 TO N%+?J%-1
25390     PRINTCHR$(1)CHR$(?K%);
25400     IF ?K%=32 AND NOT (?(K%-1)
=32) W%=W%+1
25410     IF ?K%=13 VDU 1,10
25420     NEXT
25430   IF L%>=31 THEN PRINTCHR$(1)C
HR$(12):L%=0
25440   PRINTCHR$(1)CHR$(13)
25450   N%=N%+?J%:J%=N%+2
25460   UNTIL ?(N%+3)=244
25470 VDU 3
25480 PRINT" Lines of last page- ";L
%''"Word total- "W%
25490 END
```

BEEBUG,
P.O.Box 50,
St Albans,
Herts.

"CHAINS" – a must for Model A's

*******R.G.H.*******

Have you been enthused to write a real memory gobbler as a novice, on the Model A? After 2 hours of PLOT85,X,Y I plotted and filled triangles all over the screen, constructing a "maize" for wee "Munchers" to scream around. Whoopee, it's all in there, and when run, the maze appears in 3 seconds - obviously the work of a novice! (Oh yeah?Ed.)

Now let's merge those munchers from another game with a little modification to stop them travelling through walls (IF POINT(X,Y)= colour_of_wall THEN stay). It's all in there, lets RUN. Drat. No room !! There is not enough memory remaining to use Mode 4 and all the well-chosen variable names, instead of unintelligible system integer variables. I become frantic again about an upgrade to the Model B just to save 20 cents a game!

The WELCOME TAPE !
I distinctly remember that "Biorhythms" makes you wait for a second part. CHAIN, of course !! Now I will have to split up the program again. I could have saved the effort in merging if I had Eureka'd earlier about CHAIN.

STEPS INVOLVED:

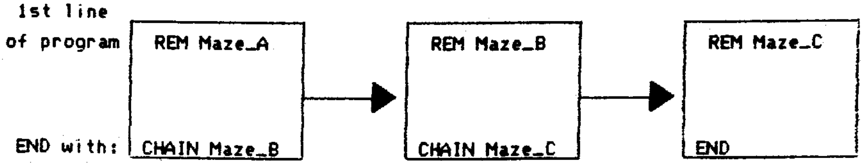
1) Save to cassette the total program which will not run, due to insufficient memory.
2) With the program still in memory, decide how you will subdivide it into smaller programs which will load and run themselves and the next ones.
   My maze program is divided in to 3 smaller programs:

   Maze_A selects the graphics mode,
   defines the graphics shapes,
   sets up the constants/start values of variables (position of munchers)
   CHAINs Maze_B.

   Maze_B Plots the maze onto the screen,
   CHAINs Maze_C.

   Maze_C Moves the munchers around the screen,
   scores and displays the Gametime,
   ends the program.

   Maze_B has since been expanded to use most of the available memory (isn't that always the way when there is plenty to spare?).
3) DELETE the line numbers you will not require for your first program. Remember, you still have the original too-long program on tape several times. Substitute the closing line of your program, which should be "END" with CHAIN "next part-program".
4) SAVE this new program "A" at the start of a fresh tape. Don't rewind after verifying, so you know where on the tape to put the next instalment. Preferably do this on at least two tapes.
5) ReLOAD your original program, and repeat the deletion process in step 3 to obtain another smaller program. Terminate it with CHAIN"3rd part" if there is more to come.
6) SAVE this part 2 a few seconds after program"A" on the tape(s).
   You can continue in this way, subdividing to eternity, until your tape starts to look like this:



A TRAP & POSSIBLE SOLUTIONS:
When using CHAIN, the title of the program and block numbers appear on the screen as with LOAD. If you have your impressive graphics laid across the screen, this is not appreciated by the Beeb, which allows these characters to chomp their way through the graphics (in my case, the Maze has holes in all the wrong places for Munchers to escape through.).
Two possible solutions:

1.) Disable the screen with VDU 21 immediately after the graphics have been plotted. Enable it again with VDU 6 just as the game is about to commence. (Using *OPT1,0 has this effect too. Ed.)
2.) Use separate text and graphics windows so that the title and block numbers are always separate from your fancy display. (Page 55 of the Users Guide covers this fully.)
    In Maze, my text window holds only 2 lines, which, when running the game, contains the Game Time, Score, and of course, "Escape" if you are having a rotten time.

Solution 1 has its drawbacks. If anything goes amiss during the loading from cassette, there is no message visible to alert you - you are well and truly in the dark !!

So, Model A owners, don't let those Model Bs show off their new fandangle memory-gobbling graphics that have "real"mountain ranges, complete with tourist resorts of hotels, swimming pools etc - just you show them there's more to being "chained" to a BBC than they ever dreamed of !

* * *

FOR THE EDITOR

Has anyone already got some routines for right-justifying of text, or for proportional spacing? As you can see, the newsletter would look much better if we could do this. Alas, preparing the newsletter takes up so much time, that we have not got around to doing this ourselves.

TO THE EDITOR

A letter was received in a recent fiery gust, from a non-member, which was rather insulting about the dragon on last month's cover. After removing the inflammatory passages, the essence was that the dr-agon had too many sides, and that no self-respecting dragon had more than two - sly-cunning, and greedy-nasty. It was signed by Smaug the Golden.

BY THE EDITOR

See p.23.

SYNCHRONISING SOUNDS

J.M.H.

While experimenting with my micro's capabilities for sound production, I ran into a problem with the Chord Synchronisation feature. This is a mechanism for ensuring that sounds on different channels start simultaneously, regardless of the earlier expiry of preceding sounds in the queues for some of the channels

The particular case I was working with involved alternately synchronising two and three channels, with some non-synchronised sounds as well. I found that the program would quickly lock up, refusing to progress at all, and it could only be freed by ESCAPE. Some experiments showed that the system was getting confused as to how many channels had to be synchronised at a time. This resulted in the wrong sounds being synchronised, as sounds that should have been executed were being held for a channel that should not have been involved. Eventually a channel queue became full of sounds waiting for synchronised entries that the program had not made; and this caused the lockup.

The particular way I was trying to work, when any two channels were being used, the third, whichever one it was, was expected to be silent. The compromise I reached was to place a very quiet sound on this channel, and making all the synchronised sounds to be three-channel. If this is not a satisfactory solution for some cases, then I suggest using the synchronising facility to a minimum, and rely instead, on the timing of sounds for simultaneous production of them.

* * *

TOWERS OF HANOI

T.B

The Towers of Hanoi is an ancient game credited to the Chinese. It's unproved origins have been dated to the Hsia dynasty (BC1800-1500). One emperor of the Ch'in dynasty is said to have challenged visitors to his court to solve the problem - then beheaded those who failed.

The traditional puzzle consists of a series of disks with holes in the centre so that they may be placed on any of three pegs. Each disk is of different diameter, so when stacked in order of size, form a cone shape.

The object of the puzzle is to move the disks one by one to transfer them all to a different pin from the one they started on. There is a rule that a disk may not have a larger one placed on top of it. This Beeb version has only six disks to manoeuvre.

There are a number of obvious enhancements to the version listed below:

1. Use the graphics capability of the Beeb to display the disks as solid disks, with different colours.
2. Show the disks actually moving from one pin to another.
3. Add appropriate sound effects.
4. Increase the number of disks.

When you tire of this version, send in your NEW AND IMPROVED version !

```
  10 REM "THE TOWERS OF HANOI"
  20
  30 DIM L1$(10)
  40 DIM L2$(10)
  50 DIM T(6,3)
  60 MODE 5:PROCtitle_1
  70 MODE 4:PROCinstruct_2
  80 VDU 23;8202;0;0;0;
  90 REPEAT
 100   PROCstart_3
 110   PROCsetup_4
 120   REPEAT
 130     MODE 4:PROCdisplay_71
 140     REPEAT PROCenter_51
 150       UNTIL E=0
 160     REPEAT PROCenter_52
 170       UNTIL E=0
 180     PROCmove_6
 190     UNTIL E$="Y" OR E$="y"
 200   PROCend_game_8
 210   UNTIL A$="N" OR A$="n"
 220 MODE 5:PROCend_run_9
 230 END
 240
 250 DEF PROCtitle_1
 260 VDU 23;8202;0;0;0;
 270 PRINT TAB(0,12) "THE TOWERS OF
HANOI"
 280 PRINT TAB(0,24) "        "
 290 TIME=0:REPEAT:UNTIL TIME=150
 300 ENDPROC
 310
 320 DEF PROCinstruct_2
 330 VDU 23;8202;0;0;0;
 340 PRINT "              TOWERS OF HA
NOI"
 350 PRINT "              ------------
---"'
 360 PRINT "      THE TOWERS OF HANO
I IS AN"
 370 PRINT "  ANCIENT PUZZLE WHOSE O
RIGINS ARE "
 380 PRINT "  LOST IN TIME."'
 390 PRINT "      THE OBJECT IS TO T
RANSFER ALL"
 400 PRINT "  THE DISKS FROM THE LEF
T HAND PIN"
 410 PRINT "  TO THE RIGHT HAND PIN.
 JUST TO"
 420 PRINT "  MAKE IT INTERESTING TH
ERE ARE A"
 430 PRINT "  FEW RULES:-"'
 440 PRINT "    1. ONLY ONE DISK MAY
 BE MOVED"
 450 PRINT "       AT A TIME"'
```

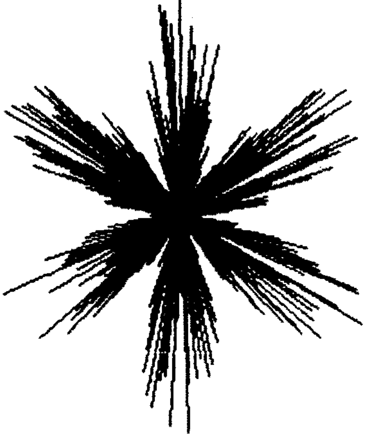
```
 460 PRINT"    2. YOU MAY NEVER PLAC
E A LARGER"
 470 PRINT "       DISK ON A SMALLER
 ONE."'
 480 PRINT "    3. DISKS MUST END UP
 ON RIGHT"
 490 PRINT "       HAND PIN NOT THE
CENTRE ONE"
 500 PRINT "      THE PINS ARE NUMBE
RED 1-3 LEFT"
 510 PRINT "  TO RIGHT. THE DISKS AR
E NUMBERED"
 520 PRINT "  BY THE NUMBER OF * S S
HOWN. THE"
 530 PRINT "  LARGEST DISK IS ALWAYS
 13."'
 540 PRINT TAB(15) "GOOD LUCK!"
 550 PRINT TAB(15) "----------"
 560 PRINT "         PRESS 'RETURN' T
O START"
 570 Z$=GET$
 580 ENDPROC
 590
 600 DEF PROCstart_3
 610 CLS:E=0:SM=0
 620 FOR D=1 TO 6
 630   FOR N=1 TO 3
 640     T(D,N)=0
 650     NEXT
 660   NEXT
 670 REPEAT
 680   PRINT TAB(5,12) "HOW MANY DIS
KS (MAXIMUM 6)?"
 690   S$=GET$:S=VAL(S$)
 700   E=0
 710   IF S<1 OR S>6 E=1:PROCerrors_
10
 720   UNTIL E=0
 730 FOR F = 1 TO S
 740   SM = (2*SM)+1
 750   NEXT F
 760 ENDPROC
 770
 780 DEF PROCsetup_4
 790 Y=6:D=13:A$="Y":E$="N"
 800 FOR X=1 TO S
 810   T(Y,1)=D:D=D-2
 820   Y=Y-1
 830   NEXT
 840 ENDPROC
 850
 860 DEF PROCenter_51
 870 PRINTTAB(8,16) "PLEASE ENTER TH
E NUMBER"
```

```
  880 PRINT TAB(6,18) "OF THE DISK YO
U WISH TO MOVE":
  890 D$=GET$:D=VAL(D$)
  900 IF D=1 THEN D$=GET$:D=10+VAL(D$
)
  910 E=2:G=13
  920 FOR F = 1 TO S
  930   IF D = G THEN E=0
  940   G = G - 2
  950   NEXT F
  960 IF E=2 THEN PROCerrors_10:ENDPR
OC
  970 REM CHECK IF DISC IS BELOW ANOT
HER
  980 FOR R=1 TO 6
  990   FOR C=1 TO 3
 1000     IF T(R,C)=D C%=C:R%=R:R=6:C
=3
 1010     NEXT C
 1020   NEXT R
 1030 C=C%:R=R%
 1040 FOR Q=1 TO R STEP 1
 1050   IF T(Q,C)<D AND T(Q,C)>0 THEN
 E=3:PROCerrors_10
 1060   NEXT
 1070 ENDPROC
 1080
 1090 DEF PROCenter_52
 1100 REPEAT
 1110   E=0
 1120   PRINTTAB(8,20)"PLACE DISC ON
WHICH PIN?"
 1130   N$=GET$:N=VAL(N$)
 1140   IF N<1 OR N>3 THEN E=4:PROCer
rors_10
 1150   UNTIL E=0
 1160
 1170 REM CHECK IF DISC IS ON LARGER
 NE
 1180 E=0
 1190 FOR R=1 TO 6
 1200   IF T(R,N)<>0 AND T(R,N)<D THE
N E=5:PROCerrors_10
 1210   NEXT
 1220 ENDPROC
 1230
 1240 DEF PROCmove_6
 1250 E=0
 1260 REM MOVE DISC
 1270 FOR V=1 TO 6
 1280   FOR W=1 TO 3
 1290     IF T(V,W)=D V%=V:W%=W:V=7:W
=3
 1300     NEXT W
 1310   NEXT V
 1320 REM LOCATE SPACE ON PIN
 1330 V=V%:W=W%
 1340 T(0,0)=T(V,W)
 1350 T(V,W)=0
 1360 FOR U=1 TO 6
 1370   IF T(U,N)<>0 U%=U:N%=N:U=6:T%
=1
 1380   NEXT U
 1390 REM MOVE DISC & SET OLD LOCATIO
N TO 0
 1400 IF T%=1 THEN T%=0:U=U%:N=N%
 1410 U=U-1
 1420 T(U,N)=T(0,0)
 1430 M=M+1
 1440 IF M>128 THEN E=6:PROCerrors_10
:ENDPROC
 1450 E$="Y"
 1460 FOR R=1 TO 6
 1470   FOR C=1 TO 2
 1480     IF T(R,C)<>0 C=2:R=6:E$="N"
 1490     NEXT
 1500   NEXT
 1510 ENDPROC
 1520
 1530 DEF PROCdisplay_71
 1540 VDU 23;8202;0;0;0;
 1550 CLS:PRINT''''
 1560 FOR K=1 TO 6
 1570   Z=6
 1580   FORJ=1 TO 3
 1590     IF T(K,J)<>0 THEN PROCdispl
ay_72 ELSE PRINT TAB(Z);"*";
 1600     Z=Z+13
 1610     NEXT J
 1620   NEXT K
 1630 ENDPROC
 1640
 1650 DEF PROCdisplay_72
 1660 PRINT TAB(Z-INT(T(K,J)/2));
 1670 V=1
 1680 REPEAT
 1690   PRINT "*";
 1700   V=V+1
 1710   UNTIL V=T(K,J)+1
 1720 ENDPROC
 1730
 1740 DEF PROCend_game_8
 1750 CLS
 1760 IF E=0 THEN PRINT TAB(11,5) "CO
NGRATULATIONS!":TAB(8,9) "YOU FINISHE
D IN ";M;" MOVES";TAB(4,11) "MINIMUM
MOVES FOR ";S;" DISKS IS ";SM
```

```
 1770 PRINT TAB(8,9) "YOU FINISHED IN
 ";M;" MOVES"
 1780 PRINT TAB(4,11) "MINIMUM MOVES
FOR ";S;" DISKS IS ";SM
 1790 REPEAT
 1800   PRINT TAB(4,18) "DO YOU WANT
TO TRY AGAIN (Y/N)?"
 1810   E=0:A$=GET$
 1820   IF A$<>"Y" AND A$<>"y" AND A$
<>"N" AND A$<>"n" THEN E=7:PROCerrors
_10
 1830   UNTIL E=0
 1840 M=0
 1850 ENDPROC
 1860
 1870 DEF PROCend_run_9
 1880 PRINT TAB(6,12) "THANK YOU"
 1890 PRINT TAB(5,14) "FOR PLAYING"
 1900 PRINT TAB(7,18) "GOODBYE"
 1910 ENDPROC
 1920
 1930 DEF PROCerrors_10
 1940 L1$(1)="   NUMBER OF DISKS MUST
 BE   "
 1950 L2$(1)="        BETWEEN 1 AND 6"
 1960 L1$(2)=" DISK NUMBER MUST BE A
VALID "
 1970 L2$(2)=" NUMBER AS PER INSTRUCT
IONS"
 1980 L1$(3)="THAT DISK IS BELOW ANOT
HER ONE"
 1990 L2$(3)=""
 2000 L1$(4)="       PIN NUMBER MUST B
E "
 2010 L2$(4)="        BETWEEN 1 AND 3"
 2020 L1$(5)="DISK MAY ONLY BE PLACED
 ON TOP"
 2030 L2$(5)="        OF A LARGER ONE"
 2040 L1$(6)="      SORRY TOO MANY MOV
ES"
 2050 L2$(6)=""
 2060 L1$(7)="    YOU MUST ANSWER Y O
R N"
 2070 L2$(7)=""
 2080 PRINT TAB(5,24)"***************
***************"
 2090 PRINT TAB(5,25) L1$(E)
 2100 PRINT TAB(5,26) L2$(E)
 2110 PRINT TAB(5,27)"***************
***************"
 2120 PRINT TAB(5,29)"  PRESS 'RETURN
' TO CONTINUE"
 2130 Z$=GET$
 2140 L$="
   "
 2150 PRINT TAB(5,24) L$
 2160 PRINT TAB(5)L$
 2170 PRINT TAB(5)L$
 2180 PRINT TAB(5)L$
 2190 PRINT TAB(5,29) L$
 2200 ENDPROC
```

Radius Function

?RND(1)*ABS(SIN(3*A))



WHITCOULLS DEALERS:

Here is a list of branches of Whitcoulls that are expected to be selling the BBC.

Auckland: Queen St
        Sunnybrae Rd
Hamilton
Tauranga
Rotorua
Gisborne
Napier
Wanganui
New Plymouth
Palmerston North
Wellington: Lambton Quay
Blenheim
Christchurch: Cashel St
        Peterborough St
Ashburton
Dunedin: Princes St
Invercargill: Dee St

USER-DEFINED KEYS

In February, we published a routine to load two sets of definitions of these keys, and be able to swap between the two sets. The User Guide does not cover their use in great detail, nor does it list the error messages you can receive when trying to define these keys ! As well, it does not make it obvious that only 256 bytes are allotted to the definitions. The suggestion of using double quotes is also confusing. Therefore, here are a few suggestions and comments to answer your queries.

ERROR MESSAGES:

Bad Key
The key definitions are allotted a limited space in memory. Consequently, when you have used up these 256 bytes, you can only redefine keys by undefining some others to give you the memory space required. By the time you have defined some of your keys with the most frequent commands - LIST, RUN, LOAD etc. if you want to start experimenting with ENVelopes say, you are likely to have space left for only three envelopes before you get that unmentionable error message, Bad Key. If you really want to experiment with long strings in these keys, you will have to think seriously about doing something like Alasdair suggested in February.

Key in use
This has occurred occasionally when trying to redefine a key. Again, it seems related to the amount of memory already used, especially if you have been redefining keys a lot. (This is more likely to happen in establishments where the computer gets so much use night and day, that no-one bothers to switch it off!) To minimise this, undefine, or clear, a key before you redefine it. The other thing to do is undefine every key and start again.

DOUBLE QUOTES or CONTROL ?

You cannot have a definition beginning with either double quotes or a comma. The computer will either tell you you have a Bad Key, or ignore the quote or comma, but produce the rest of the line. This is where the User Guide's suggestion of putting a definition in double quotes (" ") is useful.
An example of this was in the January newsletter, p.17, where to get a key defined to produce ,& on demand was enclosed in quotes. Even worse is the case of the quotes themselves - the mainstay of this newsletter production is a key defined to put in the closing quotes of a line, do a carriage return (which automatically gives the next line number), and start the new line with PRINT".
To do this required 3 sets of quotes at each end !
Since those early days, the use of the broken line (control) (¦) has eased this in that instead of counting quote marks, you need only put the control character at the beginning of the definition. ¦,& gives a ,& when pressed, which is used a lot in setting up character definitions or data in hex. *KEY2¦"¦MP." gives the required definition instead of *KEY2"""¦MP.""" .(And to put this in a PRINT statement requires 6 sets of quotes at each end to come out right !)

OTHER CONTROLS

¦U clears the line you are on. Normally if you have a key defined to LIST, and press it in the middle of a line, you will get 'Mistake'. If you start the LIST definition with a ¦U that will clear the line and start the definition at the beginning of that same line.
Alternatively, you may prefer to start your definition with a ¦M or carriage return - this will start your definition on the next line instead.

!N puts you in paging mode so you can step through listings a screenful at a time, using SHIFT key to pass on to the next page. To stop and edit on a page, you need to Escape. Note that you are still in paging mode ! If you do not do a Control O (ie press these keys simultaneously) at some stage, you will find the computer hangs up in the most awkward places:- if you return to writing a program using AUTO, periodically you will have to press Shift to get the rest of the line you have entered (usually happens as you reach the end of one line and press Return - to find a little red light under your hand!). The other popular place for the computer to get upset about being left in paging mode is when you try to SAVE or *LOAD 8000. The Red Shift light goes on, and to Escape is your task.

BREAK

Preschoolers in particular have a predilection for this key, along with the Escape key. Whilst use of Escape is annoying at the wrong moment, Break is much worse. It certainly pays to redefine the Break key so that if itchy fingers jump in you have a better chance to recover. Break is key 10, and OLD is the command to recover a program after use of Break:

*KEY10OLD!M - this is under the 0.1 OS - apparently to Break under the series 1 OSs, you have to do a Control Break to break out. Even with the Break key so defined, pressing it twice quickly will still clobber the machine.

DEFINED KEYS = INKEY

On the welcome tape, the sketch program defines some of these keys to change the colours. When you have finished with this program, the keys are still defined so that if you wish to start work on another program, you have to set up your favourite definitions again. Instead of defining the keys in the Sketch program, the programmer could have used INKEY to check which key had been pressed, then taken action accordingly. This would have left previously-made definitions undisturbed.

WHY USE THEM?

1. To cut down time spent typing in the same phrase or command during development of a program: 'The BBC Microcomputer Users Group of NZ' is a lot to type in every few lines: pressing one key with this phrase inside it is much quicker. Doing a lot of drawing? *KEY3MOVE *KEY4:DRAW to save time.
2. Of course you are SAVEing your programs twice and verifying them, even if duplicate copies are on the same tape. Why type in the whole SAVE procedure twice, with a wait between each save? The computer has a buffer, so it can remember if you give it a command before it has finished executing a previous command. *KEY3SAVE"name"!M!M . Press f3 twice, with your cassette ready to record, and go and make that cup of coffee while the machine does all your work twice! (or more) in succession.
3. By all means use these keys instead of black ones for a game. Warning: people enthousiastically hitting f9 so shift a frog across the road in a hurry are likely to hit Break - TWICE
4. Mummy, I want green letters ... Mummy, I want purple letters ... and, of course, he wants them to be big letters, which is the 4-colour mode 5. *KEY3!UMO.5:V.19.3.2.0.0.0!M and *KEY4!UMO.5:V.19.3.5.0,0,0!M and so on. Let the child remember which is which.

WHAT DOES THIS KEY DO ?

One way to find out is by typing in AUTO then pressing the key to get a listing of its definition. Of course, if you have a program already in the machine that you do not want to corrupt, then you will enter AUTO5000 instead. Or some safely high line number that the program does not reach to.

Or, you can slip a card of piece of paper under the clear perspex (?) strip that is above the red keys, with each definition written in the right place. Computer cards are just the right length for the 9 keys, and stick up beyond the perspex so you can pencil in temporary definitions easily.

ENTERING DEFINITIONS:

You can enter each definition manually, in immediate mode, or, as the User Guide suggests, write a program, save it to cassette (you've got disc?) and CHAIN the cassette each time you turn the machine on. *Below is a documented* program to define the keys - the listing also prints to screen how the keys have been defined, in case you've forgotten. If you are using the cassette bug fix, this can be incorporated in the same program. Note that after running this program you should type in NEW before you start work on another program.
You need not use these definitions, but substitute your own wherever you like.

```
 10 MODE3
 20 PRINT"Defining function keys"
 30
 40
 50 PRINT'"Reset all function keys 'f0' to 'f10'"
 60 *KEY0
 70 *KEY1
 80 *KEY2
 90 *KEY3
100 *KEY4
110 *KEY5
120 *KEY6
130 *KEY7
140 *KEY8
150 *KEY9
160 *KEY10
170
180 PRINT'" 'f0' pretty list in page mode"
190 *KEY0¦UCL.¦ML.07¦MMO.3¦M¦NL.¦M¦O
200
210 PRINT'" 'f1' list specified range of lines'
220 *KEY1¦UCL.¦ML.00¦MMO.3¦MLIST
230
240 PRINT'" 'f2.3.4,5' to redefine  *KEY2.3.4.5"
250 *KEY2*KEY2
260 *KEY3*KEY3
270 *KEY4*KEY4
280 *KEY5*KEY5
290
300 PRINT'" 'f6' turn cassette motor off"
310 *KEY6¦U*MOTOR0¦M
320
330 PRINT'" 'f7' turn cassette motor on"
340 *KEY7¦U*MOTOR1¦M
350
360 PRINT'" 'f8' load next program from cassette"
370 *KEY8¦ULO.""¦M
380
390 PRINT'" 'f9' run current program"
400 *KEY9¦URUN¦M
```

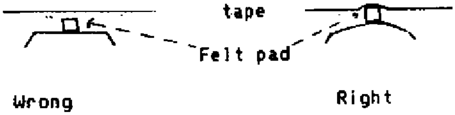
```
410
420 PRINT'" 'f10' make 'BREAK' key 'safe'"
430 *KEY10OLD¦M
440
450 PRINT'  "KEYS now defined"
460 END
```

Extended Character set for C.ITOH 5810A printer

128	&80	▁	129	&81	▂	130	&82	▃	131	&83	▄
132	&84	▅	133	&85	▆	134	&86	▇	135	&87	█
136	&88	▏	137	&89	▎	138	&8A	▍	139	&8B	▌
140	&8C	▋	141	&8D	▊	142	&8E	▉	143	&8F	┼
144	&90	┴	145	&91	┬	146	&92	┤	147	&93	├
148	&94	▔	149	&95	─	150	&96	│	151	&97	▕
152	&98	┌	153	&99	┐	154	&9A	└	155	&9B	┘
156	&9C	╭	157	&9D	╮	158	&9E	╰	159	&9F	╯
160	&A0	α	161	&A1	β	162	&A2	γ	163	&A3	δ
164	&A4	ε	165	&A5	ζ	166	&A6	η	167	&A7	θ
168	&A8	ι	169	&A9	κ	170	&AA	λ	171	&AB	μ
172	&AC	ν	173	&AD	ξ	174	&AE	ο	175	&AF	π
176	&B0	ρ	177	&B1	σ	178	&B2	τ	179	&B3	υ
180	&B4	φ	181	&B5	χ	182	&B6	ψ	183	&B7	ω
184	&B8	Δ	185	&B9	Γ	186	&BA	Σ	187	&BB	Λ
188	&BC	Ω	189	&BD	Φ	190	&BE	Ψ	191	&BF	•
192	&C0	↑	193	&C1	↓	194	&C2	←	195	&C3	→
196	&C4	±	197	&C5	≠	198	&C6	≥	199	&C7	≤
200	&C8	≈	201	&C9	·	202	&CA	⊕	203	&CB	∞
204	&CC	∴	205	&CD	½	206	&CE	¼	207	&CF	°
208	&D0	¹	209	&D1	²	210	&D2	³	211	&D3	⁴
212	&D4	⁵	213	&D5	⁶	214	&D6	⁷	215	&D7	⁸
216	&D8	⁹	217	&D9	⁽	218	&DA	⁾	219	&DB	⁺
220	&DC	⁻	221	&DD	·	222	&DE	*	223	&DF	⁄
224	&E0	=	225	&E1	╞	226	&E2	╪	227	&E3	╡
228	&E4	◢	229	&E5	◣	230	&E6	◥	231	&E7	◤
232	&E8	♠	233	&E9	♥	234	&EA	♦	235	&EB	♣
236	&EC	●	237	&ED	○	238	&EE	╱	239	&EF	╲
240	&F0	╳	241	&F1	┄	242	&F2	┆	243	&F3	∸
244	&F4	÷	245	&F5	┤	246	&F6	├	247	&F7	┼
248	&F8		249	&F9		250	&FA		251	&FB	
252	&FC		253	&FD		254	&FE		255	&FF	

TAPE TIP:
After having some loading and saving difficulties with my last batch of tapes, I found the following cure: apply some pressure to the tape's felt pressure pad so that it pushes the tape against the heads properly. Mine all suffered from a lack of tension in the small metal spring which carries out this function.

A.A.



SKETCH LOGIC REWRITTEN

G.C.F.

MOTIVATION:
All the programs on the 'WELCOME' tape are designed to run on a Model A, which has only 16K RAM. This imposes restrictions on the graphics modes available. The Sketch program runs in Mode 5 which means that the vertical lines are twice as thick as horizontal lines, and I decided to change this to Mode 1. There were also some other enhancements I wanted to make, which meant looking at the existing code, ... and understanding it !

EXISTING LOGIC
I found some incredibly complex logic to implement the fairly simple functions. It was riddled with GOTOs, and redefined the User-definable Keys unnecessarily. This logic cried out for rewriting - besides which, what did:

`*KEY 0 |!|A` do ? Answer: see p. 142 of the Users Guide: `|!|A=CHR$(128+1)`

And it was only when I had rewritten the logic from first principles, that I fully understood how they did things !
I have only rewritten that part of the logic which is required for its main function of drawing pictures, which is contained in lines 110 - 530 and 1100 - 1299 . Although, to be complete, the program needs to display a title and the instructions. Nevertheless, the listing given here works in itself.

CHANGES
The rewritten logic:-

1. Reduces the number of statements from 57 to 41.
2. Does not use GOTOs.
3. Does not redefine the User-definable Keys.
4. Allows diagonal lines to be drawn by holding down two cursor keys at once.
5. Is more easily understood in terms of what it actually does; so that enhancements can be made with less chance of unwanted side effects.
6. Still allows the User to do exactly what s/he did before, and in the same way.

* * *

WHAT OTHERS ARE DOING:

Report from one member, who converted the C.ITOH screen dump from the February issue to run on his IBM printer (Epson MX80!). He claims it took about 20 minutes to run!

Another is thinking of producing a WP which would print in Chinese characters, perhaps involving User-defineable characters. He expects this to be a project taking several years.

On graphics, one potential customer asked the Beeb salesman why the Beeb didn't have the graphics characters printed on the keys like the ATARI does! The response was that the Beeb didn't have enough keys to put them all on!

Two members report that they have imported their own Beebs for $1100 including Sales Tax.

Another has his Beeb connected to a CP/M system, but it is not yet a complete implementation.

The Users Group President prefers half a Beeb to a whole Spectrum. (He part-owns one with the Secretary, who in term also has a half-share in a Microbee!)

```
 10 REM "Sketch" logic rewritten
 20 REM G.C.F. 17-APR-83
 30 MODE 1
 40 VDU23;11,0;0;0;0
 50 VDU19,3,6;0;
 60 VDU19,2,3;0;
 70 C%=1
 80 X%=639
 90 Y%=511
100 REPEAT
110   IF INKEY(-58) AND Y%<1020 THEN Y%=Y%+4
120   IF INKEY(-42) AND Y%>3 THEN Y%=Y%-4
130   IF INKEY(-26) AND X%>3 THEN X%=X%-4
140   IF INKEY(-122) AND X%<1276 THEN X%=X%+4
150   GCOL 4,0
160   FOR I%=1 TO 2
170     MOVE X%,Y%-12
180     DRAW X%,Y%+12
190     MOVE X%-12,Y%
200     DRAW X%+12,Y%
210     NEXT I%
220   IF INKEY(-33) THEN C%=1
230   IF INKEY(-114) THEN C%=2
240   IF INKEY(-115) THEN C%=3
250   IF INKEY(-21) THEN C%=-1
260   IF INKEY(-120) THEN C%=0
270   IF C%>-1 THEN GCOL 0,C%:PLOT 69,X%,Y%
280   UNTIL FALSE
```

PRIME CHALLENGE

G.C.F.

PROBLEM:
Write a program to list all prime numbers less than 10,000 in BASIC (without using Peeks & Pokes), and see if you can get it done in less than 235.4 seconds (about 3.5 minutes). Meaningful variable names are not required ! You may assume that the first three primes are 2, 3, & 5, and the use of the built-in clock (TIME) to time the program's run is permissable.
More enterprising readers may write a machine code program to do it, but I am more interested in the methods used, and their implementations in BASIC

BACKGROUND:
There are several different methods that can be used and it would be interesting to compare their run times.
One particular method was covered in "BYTE" magazine (Jan '83) in an article titled "Eratosthenes Revisited: Once More through the Sieve". Comparative timings are given for many machines and several languages (eg Apple II, using integer BASIC - 30 minutes) also the checked numbers up to 16,381 (BBC BASIC lists these in 7 minutes).

SOLUTIONS:
If anyone can devise a program which is either significantly faster (5%plus) or more elegant than the solution I have, please send it in for publication.

PENNY PINCHING PRINTER PROJECT - PT 1

W.W.

Don't despair if you think that a printer is beyond the reach of your budget. You may have dreamt of daisy wheels, dot matrices and pen plotters all singing and dancing around your Beeb as they came to take you away to debtors prison. Forget all that. You can have your own printer at a price that is hard to beat. For less than $50 all up you can have your very own plain paper printer. It has some limitations, mind you, but it is a printer, and at that price, well...... Now read on.

The march of new technological development leaves in its wake a trail of obsolete equipment. This is usually still in good working order, not really worn out. It merely lacks the facilities which are required for today. Take the NZPO teleprinter for example. The older, mechanical Creed models have been superceded by Olivetti and Sagem machines which incorporate some of the fruit of the modern microelectronic age. These latter incorporate such enhanced features that there is no longer a place for the Creed in the Telex customers offices. Ex Post Office Creed model 54 teleprinters are available at $10 each, and it is these machines that this printer project is about.

You should be aware, from the outset, that the teleprinter has its limitations. The main ones are:- size and weight, noise, slowness, limitations of the character set and maintenance difficulties. The size, weight and noise mean simply that you put it in the garage and connect it via a cable to the BBC in your lounge.

The printing speed is 50 baud, which translates to about 6 characters per second at full steam. However, the software routine, even in machine code, takes a few milliseconds to sort out and translate each character, so the speed actually achieved is a bit less than this.

The character set is limited to upper case letters, numerals, and a reasonably sparse set of punctuation and mathematical symbols. This is not too much of a problem in most applications, but it does mean that a fair bit of handwork is required to mark up listings. Also you need to watch for traps where both lower and upper case letters come out in upper case.

The whole project involves three main steps:-

1. Obtain your teleprinter.
2. Construct an interface to boost the signal from the computer.
3. Load a software routine that will cause the required codes to be sent.

This month we look at how you get your teleprinter.

Surplus teleprinters are normally only sold by the Post Office to its own staff members, but the Users Group, however, has a fairly strong contingent of these. If you send an S.A.E. to The BBC Microcomputer Users Group of NZ, Dept TX, PO Box 9592, Wellington, together with $10 ,we will send you full details of when and where to uplift your machine (note that because of the size and weight the machine must be collected from Wellington by the purchaser).

You can be pretty sure that the machine you get was working quite happily up to the time it was taken out of service. There is, however, no guarantee, given either by the Users Group or by the Post Office. The first thing to do is plug the power cord from the teleprinter into the 3-pin socket on the back of the control unit. Next plug the multiway cable from the teleprinter into the multiway socket on the back of the control unit. Finally plug the power cord from the control unit into the mains and switch on. Press the LOCAL button on the control unit, and you should now have an electric typewriter. Pounding a few keys should print up in red on the paper (the machine prints red for transmitted text and black for received text). If you are out of roll paper or ribbon then refer to the Post Office's 'Supplement to the Telex Directory', a small orange booklet which includes a chapter on material supplies and how to obtain them from Post Office Supply Branches.

If your teleprinter does not operate as it should then you should try to make the acquaintance of a Post Office telegraph technician. We recommend that you don't meddle with the innards of the beast unless you know exactly what you are doing, and even then only make one adjustment at a time. Don't open the case while the machine is still plugged into the mains.

Next month we will look at the interface that you will need to build, and also the software driver.

TECHNICAL AIDS FOR HANDICAPPED PEOPLE

Neil SCOTT

We have been asked to pass the following on to our members, by The N.Z.Computer Society, Wellington Branch. It may be of interest to some of you - we had a program in a recent issue of our newsletter to increase the size of text on a screen, written with partially-sighted people in mind. (Ed)

A new "Technical Aids" trust has been established specifically to assist with the provision of technical aids for handicapped people. The trust provides a framework within which a wide range of volunteer skills can be coordinated. Because computer techniques play a major part in the provision of these aids, the Computer Society has a member on the board of trustees. As well, several other trustees are also members of the Society.

There is a large amount of computer hardware that could be adapted for use by the handicapped but it will need a great deal of specially written software. Some of this is reasonably straightforward, but as well, there are problems that are being tackled by some of the foremost researchers in the artificial intelligence community. One of the most exciting aspects of working in this area is that it provides a common ground for exploration by members of vastly different disciplines. For example, we have linguists, programmers, therapists and engineers all working together to find a solution to one particular communication problem.

So if any of you would like to become involved in some aspect of providing technical aids for the handicapped in either hardware or software, please contact Neil Scott or Paul Bryant at Wellington Polytechnic (ph.850-559or Private Bag, Wellington), or Bill Williams at the National Office of the Computer Society (ph.727-421 or P.O.Box 2788, Wellington).

There will be a short familiarisation course offered in June as an introduction to this field, for those interested. It will look at:

The special needs of the handicapped;
Familiarisation with the equipment & programmes that are being used;
Definition of particular problems that need to be investigated;
Possible hardware & software solutions to the defined problems.

Course details: 4 Monday evenings: June 13,20,27, & July 4
Time: 5.30pm - 8.30pm
Venue: Room 3D26, Wellington Polytechnic
Cost: $12.00

POLAR PLOTTING

G.C.F.

PREAMBLE

The first draft of this article and program was solely concerned with demonstrating the techniques of polar plotting, and to provide a couple of useful PROCedures to do this. But PROCs need a "mainline" to make them into a program that can be run to actually show what Polar Plotting can do! And that was where the trouble started. People who saw the program complained that it was NOT 'USER FRIENDLY' - a CRUEL and UNDERHAND blow to my PROFESSIONAL PRIDE. (Professionsal pride: a weakness of people who, by claiming especial competence in a particular skill, feel compelled to remedy any justified criticism. For a professional Analyst or Programmer, to be accused of not being USER Friendly is like a priest being accused of not being respectful to his GOD.)
I hope the resulting program goes some way towards dispelling such criticism, and I beg forgiveness ...

Although the main purpose of this article is to show how to do Polar Plotting on the Beeb, there are some additional comments on USER FRIENDLINESS, and BASIC technicalities. (What about the icebergs? -Ed)

INTRODUCTION

Polar Plotting can be used both for fun graphics, and also engineering applications, besides being useful in the educational context of teaching differences between Polar and Cartesian coordinate systems.
Two different PROCs are given here. The first will simply give an outline curve and the second provides a filled-in shape with random colours.
The program itself not only shows how the PROCs can be used, but also allows you to easily experiment with different functions, and to show how small modifications change the generated pattern.

MATHEMATICS

A 'Polar Plot' is the name given to a graph drawn using the 'R' (radius) and'θ' (angle) system instead of the more common rectangular 'X' and 'Y' coordinates.
In general, the radius (R) is expressed as a function of the angle(θ) eg:

R=(1+cos(θ)/2

Polar Plotting is often preferred when it is easier, or more meaningful, to define the equation for the graph in terms of R,θ rather than X,Y terms.

INSTRUCTIONS

(a) The first thing the program requires to know is which type of polar plot the User wants.
(b) It then asks for the 'Radial Function' - type in one of the examples given below, or your own choice. Note that, next time around, the previous function is redisplayed, so you can reuse parts of it to construct the next one, using the cursor keys.
(c) Thirdly, a prompt appears after the pattern has been completed, asking if you want another plot. A 'Y' answer returns to step (b).
(d) A reply of 'N' to the above prompt leads to another prompt, which gives you a chance to end the program or return to step (a).

(e) At any stage, even while plotting, pressing <ESCAPE> will return you to (a)
(f) Any errors detected by the system will be displayed and the program will return you to step (b).

USER FRIENDLINESS

I do not claim that the program is as 'USER FRIENDLY' as could be possible, but it does have some of the necessary features:-

(a) Use is made of the 'ON ERROR' and 'REPORT' features to help USERS determine why the program did not like the radial function entered.
(b) The ESCAPE key will abandon any plot without waiting for its completion, whilst still remaining in the program. This feature is useful if you selected the wrong type of plot, or don't like the current radial function.
(c) The current or last radial function is always displayed. This is especially useful in error situations, or when you are investigating ones which are very similar. Also, it is useful to have a reminder of how the pattern was generated.
(d) A graphics window is set up to ensure that the radial function expression is never overwritten by the pattern.

USER NOTES

(a) To end the program either follow instructions and reply 'Y' to the prompt 'End Program (Y/N) ?' or press <BREAK>
(b) As the keyboard does not have the Greek letter 'θ', the program uses 'A' instead.
(c) The program supplies its own scaling factor (480) so that it is best to ensure that the maximum value of the radial function does not exceed one.
(d) Negative values of 'R' have an obvious effect !

DESIGN NOTES

(a) designed to be 'USER FRIENDLY' as above.
(b) PROCs are 'self-contained' and can be incorporated easily into other programs.
(c) The PROCs are also well-behaved, and restore the graphics window and origin to the default values on exit.
(d) The PROCs allow the programmer/USER to think in terms of R&θ although they actually PLOT (or DRAW) in terms of x&y coördinates; using the transformation   x=Rcosθ
    and   y=Rsinθ

CODING NOTES

(a) The radial function to be plotted is passed as a string variable parameter (F$) from the mainline to the plotting PROCedure.
(b) Whenever variables in a PROC have no, or possibly different, significance elsewhere, they have been declared LOCAL to prevent unwanted side-effects.
(c) The number of points plotted, and the range of angle used are set by the variables 'D' and 'H', which are in radians rather than degrees.
    (2π radians = 360 degrees)
(d) Consider the use of the EVAL feature in this program - it would have been rather more complicated without it. Care should be taken, however, in its use as error messages generated by it can be more misleading than when expressions are specified directly. eg compare:

```
            10 R=EVAL("COS(A) 4")
and         10 R=COS(A) 4
```

(e) If the graphics origin is redefined (VDU29) and a graphics window is to be set up (VDU24) then they must be specified in this order.

(f) The above may help explain the negative values used, in the PROCs, for the VDU24 arguments !

(g) VDU26 cancels the effects of both VDU24 and VDU29, which was found most necessary in the version which incorporated a screen dump !
(What it did to my poor printer's ribbon ! Sucked it dry with long black lines. - ED.)

(h) DEF FN_exit(A$) has NOT been included in the listing - see p.22 of the April issue to copy it from - this newsletter is full up!

SAMPLE FUNCTIONS TO TRY:

1. SIN(A)
2. SIN(A+PI/4)
3. ABS(SIN(A))
4. (1+SIN(A))/2
5. RND(3)/3
6. ABS(SIN(2*A)+SIN(3*A))/2
7. ABS(SIN(2*A)+3*SIN(3*A))/4
8. ABS(SIN(A)+SIN(2*A))/2
9. ABS(COS(2*A))/(ABS(COS(4*A))+1)
10. ABS(COS(A)*SIN(3*A))

```
   10 REM Program to demonstrate Polar Plots
   20 MODE 1
   30 error%=FALSE
   40 F$=""
   50 ON ERROR REPORT:IF ERR=17 THEN error%=FALSE ELSE error%=TRUE:PRINT''"Please
try again"
   60 IF ERR=26 THEN PRINT'"Only variable is 'A',"'"which is the angle in radians"
   70 REPEAT
   80   IF NOT error% THEN INPUT ''"1 - Outline"'"2 - Filled"',R%
   90   REPEAT
  100     PRINT TAB(1,31)SPC(LEN(F$))
  110     PRINT''"Radius Function"'" "F$
  120     INPUT F$
  130     CLS
  140     PRINT TAB(1,31)F$;
  150     IF R%=1 THEN PROC_polar_plot_1(F$) ELSE PROC_polar_plot_2(F$)
  160     exit%=NOT FN_exit("Next plot")
  170     UNTIL exit%
  180   error%=FALSE
  190   UNTIL FN_exit("End program")
  200 MODE 3
  210 END
  220
  230 DEF PROC_polar_plot_1(F$)
  240 REM Plot outline curve only
  250 LOCAL A,D,H,R
  260 H=2*PI
  270 D=H/500
  280 VDU 29,640;512;
  290 VDU 24,-480;-480;480;480;
  300 FOR A=0 TO H STEP D
  310   R=480*EVAL(F$)
  320   PLOT 69,R*COS(A),R*SIN(A)
  330   NEXT A
  340 VDU26
  350 ENDPROC
  360
  370 DEF PROC_polar_plot_2(F$)
  380 REM Plot filled in curve using random colours
  390 LOCAL A,D,H,R
  400 H=2*PI
  410 D=H/500
  420 VDU 29,640;512;
  430 VDU 24,-480;-480;480;480;
  440 FOR A=0 TO H STEP D
  450   MOVE 0,0
  460   GCOL 0,RND(3)
  470   R=480*EVAL(F$)
  480   DRAW R*COS(A),R*SIN(A)
  490   NEXT A
  500 VDU26
  510 ENDPROC
  520
  530 DEF FN_exit(A$)
```

ON FILLING THE NEWSLETTER

The Editor

WHAT TO WRITE?

Anything related to the Beeb. It does not have to be a 20-page serial on some 'obscure' aspect of the Beeb. We are trying to cover a wide range of members' interests and abilities, and this is reflected in the varying approaches taken to similar topics in different issues. A lot of you are only learning to program on the Beeb; if you have come across some curiosity, write and tell us, or if it is a stumbling block, ask for help - we will try to find someone with the knowledge necessary to write up your problem - just give plenty of details on what you were trying to do, and how you were trying to do it.

Hardware projects are also interesting to a lot of members, so please tell us what you are doing.

I CAN'T WRITE

Then type? It doesn't matter what your ability in writing is like, so long as you express yourself clearly. Take time to use sub-headings for instance, or we can go through your article or letter and rewrite it a little if it seems needed to get your message across. Material can be submitted in legible handwriting, typed, or on cassette (!) using perhaps the mini text-editor from BEEBUG in this issue, or as a program consisting entirely of PRINT statements. We will check any programs sent in, to make sure there aren't any obvious bugs, before printing them, and generally try to maintain a high standard of presentation on your behalf. If you think it necessary, send us a draft of your article for comment first.

HOW WE EDIT

The Editor has a core of sub-editors to help fill in her huge gaps of knowledge. Articles dealing with hardware, for instance, go to Warren, who has written up several items for us. He also has been having fun (?) trying out games sent in. Gavin checks the programming and responds to queries on BASIC. There is also a telephone which gets used frequently to find out all sorts of snippets.

At times then, the editor feels more that her role is to program the printer! That can be fun too. Somewhere in this issue is a list of the Greek character set that comes with our printer. If you want to use any of these characters, they are readily available. Add a note with your article to this effect when you write in. Furthermore, I can download my own special characters to the printer's RAM (all 3K of it) and print using them. Have a look at the April issue to see lots of these characters - 4 different posturing people on p.6; the treble clefs on p.9 are two characters stacked on top of each other (with an alteration to the gap between lines so that they join without showing.
I can also do screen dumps - see Area of Polygon article for hex-agon, kiwi-gon, and the dr-agon on the cover. So was the elephant. A different example of printer-programming is on p.5 - the memory map.

If you want a diagram or illustration to your literary effort, either draw it in a reproducible form, or onto the screen for me to dump off, or in the case of special characters, send in the character on graph paper or a grid for me to work off. Multiples of 8x8 squares as used by the Beeb are fine, but don't let this restrict you.

TECHNICAL BOOKS LTD

Are offering a 10% discount to our members when purchasing computer books from them. They are at 222 Lambton Quay. Please present your membership card when asking for this discount.

TAPES

We again have some stock of C-10 leaderless tapes.
They are available in multiples of 20 for $24.00; plus $2.00 p&p if required.

CLASSIFIED ADS:

Classified Ads are free to individual members of the Users Group.
For cassette leads for motor control (ie a 7-pin DIN plug connected to three jacks), either phone Kevin Wgtn 845-243, or write to him c/- P.O.Box 9592,Wgtn. They are available to individual members for $5.00 plus 50cents p&p.

USER GROUP MEETINGS

AUCKLAND:- meets 2nd Wednesday of the month, 7.30 pm; VHF Clubrooms, Hazel Ave Mt Roskill. Ph. Dave 603-396 or Kerry 695-355 for more details.

WELLINGTON:- meets fourth Thursday of the month, 7.30 pm; at the Correspondence School, 1st floor, Staffroom. CONTACT: Warren 787-005 or Anton 286-289 if further details are required.

OTHER CENTRES:- let us know, so we can publish relevant details here.

The NEWSLETTER:

NEXT MONTH:-
- Prime numbers
- Numberbet - colours and counting for preschoolers
- Lissajous
- More on the teleprinter conversion
- What's on the Menu?
- .... and more ....

CONTRIBUTIONS:- are most welcome, in all shapes and sizes, and on any related topic. Program listings should be submitted on cassette, to avoid any copying errors. Include on paper an explanation of it,etc. Please include stamps for return postage of your tape. Please SAVE your tape program twice - the second time at at 300 baud if possible, and verify them.

DEADLINES:- Material for a particular issue should be with us by the last day of the month prior to the month of publication.

ADVERTISING:- Rates are $20.00 per half page.

Copyright 1983 The BBC Microcomputer Users Group of NZ, who publish this newsletter monthly. It is available to financial members.

MEMBERSHIP:

Membership of the Group is on payment of an annual subscription from April to March. For the 1983/4 year it has been set at $20.00, plus there is a joining fee of $5.00 for new members. New members will be sent all back issues of the monthly newsletter, which first came out in January 1983.
Local meeting groups can add an extra levy to cover room etc costs; for both Auckland and Wellington, this has been set at $5.00.

THE BBC MICROCOMPUTER USERS GROUP OF NZ
P.O.BOX 9592,
WELLINGTON.